We Claim:

1.   A method of selectively or indiscriminately blocking access to a volume at least a portion of which is on a storage-device that has been formatted according to a predetermined file-system, the method comprising:

preventing, selectively or indiscriminately, a file-system driver that corresponds to the predetermined file-system from completing a mount-request.

2.   The method of claim 1, further comprising:
carrying out the preventing step at the volume-level.

3.   The method of claim 1, further comprising:
preventing, selectively or indiscriminately, the file-system driver from attaching a device object (DO) to a stack of DOs representing the volume.

4.   The method of claim 3, further comprising:
attaching a blocking filter device object (DO) at a location in the stack where otherwise a DO of a file-system driver corresponding to the predetermined file-system would have been attached.

5.   The method of claim 4, further comprising:
using, selectively or indiscriminately, the blocking filter DO to fail an input/output request packet (IRP) whose downward traversal through the stack reaches the blocking filter DO.

6.   The method of claim 3, further comprising:
selectively skipping the preventing step according to an identifier (ID) of the volume (volume-ID).

7.   The method of claim 6, further comprising:

determining if the volume-ID is present on a list of volumes for which access is to be blocked; and

doing one of

permitting, if the volume-ID is not one for which access is to be blocked, a file-system driver to claim ownership of the volume, and

posing, if the volume-ID is one for which access is to be blocked, as a file-system driver to claim ownership of the volume.

8.   The method of claim 3, further comprising:

obtaining an identifier (ID) of the volume (volume-ID) from a file-system driver corresponding to the predetermined file-system before completion of a mount-request made to the file-system driver.

9.   The method of claim 8, wherein volume-ID is obtained before the I/O manager receives indication that the file-system driver has claimed ownership of the volume.

10.   The method of claim 8, further comprising:

denying the mount-request, despite otherwise successful progress toward successful completion, if the volume-ID is one for which access is to be blocked.

11.   The method of claim 3, further comprising:

switching input/output (I/O) states according to at least one of a first transition and a second transition;

the first transition going from a prevented-state in which the file-system driver has been prevented from mounting the volume to a permitted-state in which the file-system driver is permitted to mount the volume;

the second transition going from the permitted state to the prevented state.

12. The method of claim 11, further comprising:

receiving an input/output request packet (IRP);

changing I/O states based upon a mode-switching (MS) control-code included in the IRP.

13. The method of claim 12, further comprising:

failing, in an active mode, an IRP that does not contain an MS control-code.

14. The method of claim 12, further comprising:

incompletely tearing down, and then rebuilding, the stack in order to achieve either the first or second transition.

15. A method, for a Windows®-type operating system, of controlling access to a volume at least a portion of which is on a storage-device that has been formatted according to a predetermined file-system, the method comprising:

providing a blocking filter driver that poses before an input/output (I/O) manager as a file-system driver that corresponds to the predetermined file-system; and

manipulating a driver load process such that a mount request made by the I/O manager goes to the blocking filter driver before a mount-request by the I/O manager can reach the file-system driver.

16. The method of claim 15, further comprising:

causing the I/O manager to make the mount-request to the blocking filter driver before the I/O manager makes any mount-request to any loaded file-system driver.

17. The method of claim 15, further comprising:

the blocking filter driver answering the mount-request by claiming ownership of the predetermined file-system.

18.   The method of claim 17, further comprising:

adding a new group name into a registry representing an order in which drivers are loaded;

locating the group name in the order registry after a name representing one or more file system drivers;

and setting an entry in a registry representing the blocking filter driver to indicate the new group name.

19.   The method of claim 15, wherein:

the blocking filter driver is not a file-system driver;

the method further comprising

the blocking filter driver posing as a file-system driver.

20.   A machine-readable medium including instructions execution of which by a machine selectively or indiscriminately blocks access to a volume at least a portion of which is on a storage-device that has been formatted according to a predetermined file-system, the machine-readable instructions comprising:

a code segment that prevents, selectively or indiscriminately, a file-system driver that corresponds to the predetermined file-system from completing a mount-request.

21.   An apparatus for selectively or indiscriminately blocking access to a volume at least a portion of which is on a storage-device that has been formatted according to a predetermined file-system, the apparatus comprising:

a memory in which is created the stack of device objects representing a storage-device, the stack including a filter device object (DO); and

filter driver means for posing before an input/output (I/O) manager as a file-system driver that corresponds to the predetermined file-system in order to prevent, selectively or indiscriminately, a file-system driver that corresponds to the predetermined file-system from completing a mount-request.

22.   The apparatus of claim 21, wherein the filter driver means is further operable for intercepting a mount request made by the I/O manager and otherwise intended for the file-system driver.

< Remainder of Page Intentionally Left Blank >